

**AMENDMENTS TO THE CLAIMS**

This Listing of Claims will replace all prior versions, and listings, of claims in this application:

**Listing of Claims:**

1.      (Currently Amended) A method for processing an input document encoded in an extensible ~~human-friendly~~ extensible markup language ("XML"), said method comprising ~~the steps of:~~

    (a)——converting said input document encoded in XML to an output document encoded in a machine-oriented extensible markup language ("mXML"), said output document encoded in mXML being capable of being processed more efficiently than said input document encoded in XML;

    (b)——processing said output document encoded in mXML;

    (c)——identifying a target to which the processed output document encoded in mXML will be next routed; and

    (d)——determining whether said target is capable of processing documents encoded in mXML;

if said target is determined to be capable of processing documents encoded in mXML, transmitting the processed output document encoded in mXML to said target;  
and

if said target is determined to be not capable of processing documents encoded in mXML, converting the processed output document encoded in mXML to an output document encoded in XML, and transmitting said output document encoded in XML to

said target.

2.      (Currently Amended) The method of claim 1, wherein ~~step (c)~~ said identifying comprises parsing the processed output document.

3.      (Currently Amended) The method of claim 2, wherein ~~step (c)~~ said identifying comprises identifying a host name string for routing of the processed output document.

4.      (Currently Amended) The method of claim 1, wherein ~~step (d)~~ said determining comprises referencing a datastore, said datastore storing data identifying a plurality of targets and indicating whether each of said plurality of targets is capable of processing documents encoded in mXML.

5.      (Currently Amended) The method of claim 1, wherein ~~step (a)~~ said converting comprises ~~the steps of:~~

{a1}—creating a document tree representation of the input document;

{a2}—obtaining a node count representing a count of nodes in the document tree representation;

{a3}—writing the node count to an mXML buffer;

{a4}—traversing each node in the document tree representation and generating a corresponding node specification in the mXML buffer, further comprising the steps of:  
generating a node name;

generating an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node;

generating a child list specifying index values of zero or more nodes which are children of the node; and

generating a node value specification, which is empty if the node has no value;

(a5)—generating a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications; and

(a6)—appending the data buffer to the mXML buffer to form the output document.

6.      (Currently Amended) The method of claim 1, wherein ~~step (b)~~ said processing comprises ~~the step of:~~

(b1)—parsing said output document.

7.      (Currently Amended) The method of claim 6, wherein ~~step (b1)~~ comprises ~~the step of:~~

(b1a)—said parsing of said output document comprises extracting routing data from said output document.

8.      (Canceled).

9.      (Currently Amended) The method of claim 1, wherein ~~step (b)~~ said processing comprises ~~the step of~~:

~~(b1)~~—processing said output document for content based routing if said target is determined ~~in step (d)~~ to be capable of processing documents encoded in mXML.

10.    (Canceled).

11.    (Currently Amended) The method of claim 1 ~~10~~, further comprising the ~~step of~~:

~~(f)~~ processing the converted output document encoded in XML.

12.    (Currently Amended) The method of claim 11, wherein ~~step (f)~~ processing the converted output document encoded in XML comprises transmitting the converted output document encoded in XML to said target.

13.    (Currently Amended) The method of claim 6, wherein ~~step (b1)~~ processing said output document encoded in mXML comprises ~~the steps of~~:

~~(b1a)~~—parsing a node count representing a count of nodes in the document;

~~(b1b)~~—parsing a node specification for each of the nodes, further comprising the steps of:

parsing a node name;

parsing a child list specifying index values of zero or more

nodes which are children of the node;

        parsing an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node; and

        parsing a node value specification, which is empty if the node has no value; and

~~(b1c)~~ parsing a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications.

14.     (Currently Amended) A method for processing an input document encoded in a machine-oriented extensible markup language ("mXML"), said method comprising ~~the steps of:~~

        (a) ~~—determining whether said input document will be next routed to a target which is capable of processing documents encoded in mXML, documents encoded in mXML excluding tags that include human language words;~~

        (b) ~~—converting said input document encoded in mXML to an output document encoded in a ~~human-friendly~~ extensible markup language ("XML") if said target is determined in ~~step (a)~~ to be not capable of processing documents encoded in mXML;~~  
and

        (c) ~~—processing said output document encoded in XML.~~

15.     (Currently Amended) The method of claim 14, wherein ~~step (a)~~ said determining comprises ~~the steps of:~~

{a1}—identifying a target to which said input document will be next routed; and

{a2}—determining whether said target is capable of processing documents

encoded in mXML.

16.    (Currently Amended) The method of claim 15, wherein ~~step (a2)~~  
determining whether said target is capable of processing documents encoded in mXML  
comprises referencing a datastore, said datastore storing data identifying a plurality of  
targets and indicating whether each of said plurality of targets is capable of processing  
documents encoded in mXML.

17.    (Currently Amended) The method of claim 14, wherein ~~step (b)~~ said  
converting comprises ~~the steps of:~~

{b1}—identifying a document tree representation of the input document;

{b2}—reading a node count from an mXML buffer;

{b3}—traversing each node in the document tree representation and generating  
a corresponding node specification in the mXML buffer, further comprising the steps of:

identifying a node name;

identifying an attribute list specifying zero or more (attribute name,  
attribute value) pair references for attributes of the node;

identifying a child list specifying index values of zero or more nodes which  
are children of the node; and

identifying a node value specification, which is empty if the node has no  
value;

(b4)—generating a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications; and

(b5)—appending the data buffer to the XML buffer to form the output document.

18.    (Currently Amended) The method of claim 14, further comprising ~~the step~~ of:

(d)—processing said input document encoded in mXML.

19.    (Currently Amended) The method of claim 18, wherein ~~step (d)~~ processing said input document encoded in mXML comprises parsing said input document encoded in mXML.

20.    (Currently Amended) The method of claim 14, wherein ~~step (e)~~ determining whether said input document will be next routed to a target which is capable of processing documents encoded in mXML comprises transmitting said output documents encoded in XML.

21.    (Currently Amended) A computer program product embodied on one or more computer-readable media, the computer program product adapted for processing an input document encoded in an ~~extensible human-friendly~~ extensible markup language ("XML") and comprising:

computer-readable program code ~~for~~ configured to converting said input

document encoded in XML to an output document encoded in a machine-oriented extensible markup language ("mXML"), said output document encoded in mXML being capable of being processed more efficiently than said input document encoded in XML;

computer-readable program code ~~for~~ configured to processing said output document ~~encoded in XML;~~

computer-readable program code ~~for~~ configured to identifying a target to which the processed output document will be next routed; and

computer-readable program code ~~for~~ configured to determining whether said target is capable of processing documents encoded in mXML.

22. (Currently Amended) The computer program product of claim 21, wherein said computer-readable program code ~~for~~ configured to converting said input document encoded in ~~an extensible human friendly extensible markup language (XML)~~ to an output document encoded in ~~a machine oriented extensible markup language ("mXML")~~ comprises:

computer-readable program code ~~means for~~ configured to creating a document tree representation of the input document;

computer-readable program code ~~means for~~ configured to obtaining a node count representing a count of nodes in the document tree representation;

computer-readable program code ~~means for~~ configured to writing the node count to an mXML buffer;

computer-readable program code ~~means for~~ configured to traversing each node in the document tree representation and generating a corresponding node specification



in the mXML buffer, further comprising:

computer-readable program code ~~means for~~ configured to generateing a node name;

computer-readable program code ~~means for~~ configured to generateing an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node;

computer-readable program code ~~means for~~ configured to generateing a child list specifying index values of zero or more nodes which are children of the node;  
and

computer-readable program code ~~means for~~ configured to generateing a node value specification, which is empty if the node has no value;

computer-readable program code ~~means for~~ configured to generateing a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications; and

computer-readable program code ~~means for~~ configured to appending the data buffer to the mXML buffer to form the output document.

23. (Currently Amended) The computer program product of claim 21, wherein said computer-readable program code ~~for~~ configured to processing said output document comprises computer-readable program code for processing a document encoded in mXML comprising:

computer-readable program code ~~means for~~ configured to parseing the document, further comprising:

computer-readable program code ~~means for~~ configured to parsing a  
node count representing a count of nodes in the document;

computer-readable program code ~~means for~~ configured to parsing a  
node specification for each of the nodes, further comprising:

computer-readable program code ~~means for~~ configured to parsing  
a node name;

computer-readable program code ~~means for~~ configured to parsing  
a child list specifying index values of zero or more nodes which are children of the  
node;

computer-readable program code ~~means for~~ configured to parsing  
an attribute list specifying zero or more (attribute name, attribute value) pair references  
for attributes of the node; and

computer-readable program code ~~means for~~ configured to parsing  
a node value specification, which is empty if the node has no value; and

computer-readable program code ~~means for~~ configured to parsing a data  
buffer containing attribute names and attribute values referenced from the attribute lists  
and node values referenced from the node value specifications; and

computer-readable program code ~~means for~~ configured to using the parsed  
document as input for the processing.

24. (Currently Amended) A system for processing an input document  
encoded in an ~~extensible human friendly~~ extensible markup language ("XML"), said  
system comprising:

means for converting said input document encoded in XML to an output document encoded in a machine-oriented extensible markup language ("mXML"), said output document encoded in mXML being capable of being processed more efficiently than said input document encoded in XML;

means for processing said output document encoded in mXML;

means for identifying a target to which the processed output document will be next routed; and

means for determining whether said target is capable of processing documents encoded in mXML.

25. (Currently Amended) The system of claim 24, wherein said means for converting said output document encoded in ~~an extensible human friendly extensible markup language (XML)~~ to an output document encoded in ~~a machine oriented extensible markup language (mXML)~~ comprises:

means for creating a document tree representation of the input document;

means for obtaining a node count representing a count of nodes in the document tree representation;

means for writing the node count to an mXML buffer;

means for traversing each node in the document tree representation and generating a corresponding node specification in the mXML buffer, further comprising:

means for generating a node name;

means for generating an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node;

means for generating a child list specifying index values of zero or more

nodes which are children of the node; and

means for generating a node value specification, which is empty if the node has no value;

means for generating a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications; and

means for appending the data buffer to the mXML buffer to form the output document.

26. (Original) The system of claim 24, wherein said means for processing said output document in mXML comprises:

.means for parsing the document, further comprising:

means for parsing a node count representing a count of nodes in the document;

means for parsing a node specification for each of the nodes, further comprising:

means for parsing a node name;

means for parsing a child list specifying index values of zero or more nodes which are children of the node;

means for parsing an attribute list specifying zero or more (attribute name, attribute value) pair references for attributes of the node; and

means for parsing a node value specification, which is empty if the node has no value; and

means for parsing a data buffer containing attribute names and attribute values referenced from the attribute lists and node values referenced from the node value specifications; and

means for using the parsed document as input for the processing.

27. (Currently Amended) A method for processing an input document comprising ~~the steps of~~:

(a) ~~—determining whether said input document will be next routed to a target which is capable of processing documents encoded in a machine-oriented extensible markup language ("mXML"), said input document encoded in mXML being capable of being processed more efficiently than said input document;~~ and

(b) ~~—converting said input document to an output document encoded in a human friendly an extensible markup language ("XML") if said input document is encoded mXML and said target is not capable of processing documents encoded in mXML.~~

28. (Currently Amended) The method of claim 27, further comprising ~~the step of~~:

(c) ~~—converting an original document encoded in XML to an input document encoded in a machine-oriented extensible markup language ("mXML");~~  
~~step (c) being performed before step (a) and wherein converting an original document encoded in XML to an input document encoded in mXML occurs prior to determining whether said input document will be next routed to a target which is capable of~~

processing documents encoded in a machine-oriented extensible markup language ("mXML"), said input document encoded in mXML being capable of being processed more efficiently than said input document.

29.    (Currently Amended) The method of claim 28, wherein ~~step (a)~~ said determining comprises the steps of:

    (a1)—identifying a target to which said input document will be next routed;

    (a2)—determining whether said target is capable of processing documents encoded in mXML.

30.    (Currently Amended) The method of claim 29, wherein ~~step (a1)~~ said identifying comprises parsing said input document.

31.    (Currently Amended) The method of claim ~~34~~ 30, wherein ~~step (a2)~~ said determining comprises referencing a datastore, said datastore storing data identifying a plurality of targets and indicating whether each of said plurality of targets is capable of processing documents encoded in mXML.

32.    (New) The method of claim 1, wherein said output document is encoded in an array notation.

33.    (New) The method of claim 1, wherein said input document encoded in XML includes tags that include human language words, and wherein said output

document encoded in mXML does not include any tags that include human language words.